REMARKS/ARGUMENTS

Reconsideration of this application is respectfully requested.

Claim Rejections 35 U.S.C. § 102

The Office Action has again rejected claims 1 and 21 under 35 U.S.C. 102(b) as being anticipated by Lotito et al. in United States Patent No. 4,625,081. Applicant again respectfully disagrees.

As noted in the response filed January 20, 2004, it is established law that in order to anticipate, a reference must teach each and every feature of the claimed invention. Lotito neither teaches nor suggest any of the claimed limitations of claim 1.

The Office Action asserts that the recitation of shared memory has not been given any patentable weight because the recitation occurs in the preamble. Claim 1 is amended to specifically recite shared memory in the claimed limitations. Claim 1 now explicitly claims a method of managing execution time in a shared memory parallel processor computing environment.

Lotito teaches a system 100 consisting of from 8 to 32 processors, together with a range of controllers, peripherals and storage modules. Each processor is a fully independent, high speed 16-bit machine having a non-micro-coded architecture and **high-speed memory** (emphasis added). In spite of the position expressed by the Office Action, any person skilled in the art understands that the system 100 taught by Lotito, which can be used in environments of large numbers of peripheral devices and storage units, is not a shared memory parallel processor computing environment.

As noted in the response filed January 20, 2004, claim 1 further claims defining a plurality of process classes and assigning each process to be executed to one of the process classes. Applicant concedes that Lotito teaches defining five process classes, numbered from 0 to 4. Processes assigned to class 4 have the highest priority; those assigned to class 0 have the lowest priority. Within

each of the 5 classes, processes are ranked from 0 to 255, the highest rank being 255 (column 102, lines 54-59). However, as previously explained, Lotito does not suggest or describe the assignment of an execution time slice for each of the process classes.

Claim 1 further claims permitting a process to be executed by one of the processors without interruption until the execution time slice associated with the process class has expired. As previously noted, Lotito teaches directly away from this limitation in columns 103 and 104. For example, Lotito teaches in column 103, lines 4-7 that "after a process begins execution, various REX routines allows the process to dynamically alter its own priority or, subject to certain restrictions, that of another process. Priority is directly associated with class in accordance with Lotito. Consequently, altering the priority of a process is equivalent to changing the class of the process. As would be understood by any person skilled in the art, if Lotito's processes change their class or the class of others, then the affected processes cannot function as claimed in amended claim 1.

The Office Action acknowledges that Lotito's processes can be terminated by an external hardware interrupt. This likewise teaches directly away from the invention claimed in claim 1. Lotito's processes therefore only execute within limits defined by consecutive hardware interrupts. In contrast, and as claimed in amended claim 1, processes in accordance with the invention are executed until the time slice associated with the process class of the process expires or the process or finishes executing the process. The process is never interrupted by a hardware interrupt.

In column 103, lines 10-13 Lotito teaches that "Once given control, a process is allowed to execute until it voluntarily relinquishes control **or is interrupted by the occurrence of a hardware interrupt.**" As explained above, this teaches directly away from the claimed invention. Furthermore, as taught in column 103, lines 61-62 of Lotito, "Processes in one class are allowed to interrupt those in a lower class". Again, this teaches directly away from the invention

claimed claim 1 which does not support preemption in any form. As further taught in column 104, lines 3-8, "The servicing of a hardware interrupt can result in making dispatchable a process whose class is higher than the interrupted process. In this case, the interrupted process is suspended and the higher-class process given execution control". Thus Lotito once again teaches directly away from the invention claimed in claim 1.

The rejection of claim 1 under 35 U.S.C. 102(a) is thereby traversed.

Claim 21 claims a computing machine which performs the steps of the method of claim 1. Claim 21 is amended to incorporate the limitations of amended claim 1 and the rejection of claim 21 is traversed for reasons set forth above.

Claim Rejections 35 U.S.C. § 103

The Office Action rejected claims 2-8, 11, 12, 25-26 and 28-31 under 35 U.S.C. § 103(a) as being unpatentable over Lotito in view of Frank et al. in United States Patent No. 5,790,851.

The Office Action expresses the position that one cannot show non-obviousness by attacking references individually where the rejections are based on combinations of references. Applicant agrees.

However, when one of the references teaches directly away from the claimed invention, the combined references cannot be relied on to establish a *prima facie* case of obviousness because the combination would not lead a person of ordinary skill in the art to the claimed invention. As explained above in detail, Lotito teaches away from the invention claimed in claims 1 and 21.

As also explained in the response filed January 20, 2004, Frank teaches a method of sequencing lock call requests to an operating system in order to avoid spin lock contention within a multi-processor environment. An arbitration procedure is used to allow processes and their associated processors to perform useful work while they have a pending service request for access to shared resources within a multi-processor system environment. The arbitration procedure is implemented within a multi-processor system in which multiple processes can

simultaneously request locks that control access to shared resources such as access to resources that are globally synchronized among the many processes. Such locks are very familiar to those skilled in the art and commonly used for operations such as relational database updates.

Considering Lotito who teaches away and Frank who teaches an unrelated lock call request, it is clear that no combination of Lotito and Frank teaches or suggests the calling of a lock procedure during execution of a process to permit the process to continue to execute without interruption for a predetermined period of time after a time slice associated with the process class to which the process belongs has expired. In fact, Lotito and Frank collectively fail to teach or suggest execution time slices associated with process classes, or any method for permitting a process class to execute without interruption beyond a hardware interrupt if execution has not completed. It is therefore respectfully submitted that the combination of Lotitio and Frank would lead a person of ordinary skill in the art away from the invention claimed in claims 2-8, 11, 12, 25, 26 and 28-31. The rejection of those claims is thereby traversed.

The Office Action rejected claims 9, 10 and 27 under 35 U.S.C. 103(a) as being unpatentable over Lotito in view of Frank and further in view of Gans et al. in United States Patent No. 5,835,762.

The Office Action again responded to Applicant's arguments against the references on the grounds that the references were individually attacked. Applicant respectfully reiterates that individual references that teach away from, or are silent about claimed features of an invention cannot be combined to lead a person of ordinary skill in the art to the claimed invention. As also explained in the response filed January 20, no combination of Lotito, Frank and Gans et al. yields the invention claimed in claims 9, 10 and 27. Therefore, the statement respecting attacking references individually is not supported, and Applicant request that it be withdrawn.

As explained above, the combination of Lotito and Frank teaches away from the claimed invention. Gans et al. teaches a method and apparatus for processing electronic mail in parallel. Mail objects are processed in an electronic mail system by managing a plurality of messages in a queue. Although the electronic mail messages are processed in parallel, there is no mention of processing using a shared memory parallel processor. Gans et al. therefore represents nonanalagous art. It is therefore respectfully submitted that there is no motivation for a person of ordinary skill in the art to combine Lotito, Frank and Gans et al.. At any rate, as noted above, no combination of Lotito, Frank and Gans et al. yields the invention claimed in claims 9, 10 and 27 and the rejection of those claims is traversed.

The Office Action rejected claims 13, 14 and 22-24 as being unpatentable over Lotito in view of Elnozahy in United States Patent No. 6,421,701. As explained in the response filed January 20, Elnozahy teaches a method and system for replication support in a remote method invocation system. As taught in column 5, lines 38-46, Elnozahy teaches that his system implements deterministic and preemptive threads scheduling. Scheduler 31 allocates instructions slices on a CPU, where an instructions slice serves as a scheduling unit during which a thread executes a prespecified number of instructions, P, before it is preempted. Thus scheduling by "instruction slice" means scheduling a thread based upon a number of instructions executed, and not based upon a time of execution or a number of execution cycles. Elnozahy therefore also teaches directly away from the claimed invention.

As explained above in detail, Lotito also teaches away from the claimed invention. Consequently, the combination of Lotito and Elnozahy cannot lead a person of ordinary skill in the art to the claimed invention, because the references respectively, and when combined, teach directly away.

Lotito teaches execution time control based on priority within a window limited by hardware interrupts. Elnozahy teaches scheduling based on a

number of instructions to be executed. The combination of Lotito and Elnozahy would therefore lead a person to a system in which process time is controlled by class and the number of instructions executed, within the window limited by the consecutive hardware interrupts. The combination of Lotito and Elnozahy therefore teaches directly away from the invention claimed in claims 13, 14 and 22-24, and the rejection of claims 13, 14 and 22-24 is traversed.

Claims 15-20 and 32-36 were rejected under 35 U.S.C. as being unpatentable over Lotito in view of George et al.

As explained above, Lotito teaches away from the claimed invention.

George et al. teaches a hardware-configured operating system kernel having a parallel-searchable event queue for a multitasking processor.

The Office Action expresses the position that George et al. teaches a timer queue being used to queue processes in a wait state until a predetermined process removal time has expired, when **a variable for storing a process storage time in the timer queue** is defined, with reference to Fig. 2, element T1 201. However, the Office Action ignores certain limitations of claim 15. In particular, claim 15 claims "defining a variable for storing a time at which the next process is to be removed" -- and "re-computing a time at which a next process is to be removed from the timer queue".

In contrast, George et al. teach that "At predetermined clock-tick intervals, T_A , the linked-list of TCBs is searched". Consequently, as understood by any person skilled in art, the linked-list taught by George et al. is searched at every predetermined clock-tick interval. George et al. neither teach nor suggest a variable time that is recomputed each time a process is removed from the timer queue.

The Office Action also asserts that George et al. teach recomputing a time at which a next process is removed from the timer queue and storing the recomputed time in the variable. This is clearly an inappropriate interruption of the claim, since the claim defines the re-computed variable as being distinct from a removal time associated with each process in the timer queue. George et al. teach

recomputing the removal time, and not a variable time for removing a next process from the timer queue. Consequently, the combination of Lotito et al. and George et al. teach directly away from the claimed invention.

Furthermore, with respect to, for example, claim 16, George et al. fail to teach or suggest "computing a delay time specified by a process to be added to the queue". The Office Action assets that computing a delay time is specified by the process T_c+x+Q 114. This variable T_c+x+Q is used, however, for computing a position in which a process is to be inserted into a linked-list. T_c+x+Q is in no way associated with replacing a variable for storing a time at which a next process is to be removed from the timer queue.

For at least these reasons, the rejection of claims 15-20 is traversed.

Claims 32-36 are claims to a machine that performs the steps of the methods of claims 15-19. The rejection of claims 32-36 is therefore likewise traversed.

In view of the amendments made to claims 1, 15 and 21 and for reasons set forth above in detail, claims 1-36 pending in this application are considered to be in a condition for immediate allowance. Favourable reconsideration and early issuance of a Notice of Allowance are therefore requested.

Respectfully submitted,

By _____
Max R. Wood
Registration No.40,388
Attorney for Applicant

MRW/ma

Address: OGILVY RENAULT
1981 McGill College Avenue
Suite 1600
Montreal, QC H3A 2Y3
Canada